

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	



```

0001 0 ZTITLE 'directs output or move of TSF'
0002 0 MODULE OUTLIN ( IDENT = 'V04-000'
P 0003 0 XBLISS32C, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
0004 0 NONEXTERNAL = LONG_RELATIVE)
0005 0 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1
0032 1 ++
0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0034 1
0035 1 ABSTRACT:
0036 1
0037 1 Either saves information described by TSF or causes it to
0038 1 be output immediately.
0039 1
0040 1 ENVIRONMENT: Transportable
0041 1
0042 1 AUTHOR: R.W.Friday CREATION DATE: May, 1978
0043 1

```



OUTLIN  
V04-000

directs output or move of TSF  
Revision History

J 12  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI;1

Page 2  
(2)

45	0044	1	%SBTTL 'Revision History'
46	0045	1	MODIFIED BY:
47	0046	1	
48	0047	1	
49	0048	1	019 KAD00019 Keith Dawson 9-May-1983
50	0049	1	Remove support for .DX, .PX.
51	0050	1	
52	0051	1	018 RER00018 Ron Randall 20-Mar-1983
53	0052	1	For DSRPLUS: Added code for topnotes.
54	0053	1	
55	0054	1	017 KAD00017 Keith Dawson 20-Mar-1983
56	0055	1	Removed LN01 conditionals and all references to .BIX
57	0056	1	and .BTC files.
58	0057	1	
59	0058	1	016 KAD00016 Keith Dawson 07-Mar-1983
60	0059	1	Global edit of all modules. Updated module names, idents,
61	0060	1	copyright dates. Changed require files to BLISS library.
62	0061	1	--

OUTLIN  
V04-000

directs output or move of TSF  
Module Level Declarations

K 12  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI;1

Page 3  
(3)

```

64 0062 1 %SBTTL 'Module Level Declarations'
65 0063 1
66 0064 1 : TABLE OF CONTENTS:
67 0065 1
68 0066 1 FORWARD ROUTINE
69 0067 1 OUTLIN : NOVALUE,
70 0068 1 OUTCRG : NOVALUE,
71 0069 1 OUTJ : NOVALUE,
72 0070 1 OUTNJ : NOVALUE,
73 0071 1 OUTPAS : NOVALUE;
74 0072 1
75 0073 1 : INCLUDE FILES:
76 0074 1
77 0075 1 LIBRARY 'NXPORT:XPORT'; : XPORT Library
78 0076 1 REQUIRE 'REQ:RNODEF'; : RUNOFF variant definitions
79 0207 1
80 U 0208 1 %IF DSRPLUS %THEN
81 U 0209 1 LIBRARY 'REQ:DPLLIB'; : DSRPLUS BLISS Library
82 0210 1 %ELSE
83 0211 1 LIBRARY 'REQ:DSRLIB'; : DSR BLISS Library
84 0212 1 %FI
85 0213 1
86 0214 1
87 0215 1 : EXTERNAL REFERENCES:
88 0216 1
89 0217 1 EXTERNAL
90 0218 1 BRNOOB : $XPO IOB (),
91 0219 1 FOOREC : FOOREC_DEFINITION,
92 0220 1 FNCT : FNCT_DEFINITION,
93 0221 1 FRA : FIXED_STRING,
94 0222 1 GCA : GCA_DEFINITION,
95 0223 1 MRA : REF_FIXED_STRING,
96 0224 1 OUTOPT : OUTOPT_DEFINE,
97 0225 1 SCA : SCA_DEFINITION,
98 0226 1 TSF : TSF_DEFINITION;
99 0227 1
100 U 0228 1 %IF DSRPLUS %THEN
101 U 0229 1 EXTERNAL
102 U 0230 1 TOPNOT : TN_DEFINITION,
103 U 0231 1 TNREC : TNREC_DEFINITION;
104 U 0232 1
105 U 0233 1 EXTERNAL ROUTINE
106 U 0234 1 TNFIL;
107 0235 1 %FI
108 0236 1
109 0237 1 EXTERNAL LITERAL
110 0238 1 RNFCJL,
111 0239 1 RNFLOC;
112 0240 1
113 0241 1 EXTERNAL ROUTINE
114 0242 1 endwrd, erm, foofil, lout,
115 0243 1 putndy, puttxt, scl, unpus;
116 0244 1 !! XOUT;
117 0245 1
```

```
119 0246 1 %sbtll 'OUTLIN -- output full MRA.'
120 0247 1 GLOBAL ROUTINE outlin (justify) : NOVALUE =
121 0248 1
122 0249 1 ++
123 0250 1 FUNCTIONAL DESCRIPTION:
124 0251 1
125 0252 1 OUTLIN is called when MRA is full, and the text should be output. It
126 0253 1 routes normal text lines directly to LOUT, for output. Other records
127 0254 1 are passed to other processors. For example, footnote records get
128 0255 1 saved in the footnote file, topnote records get saved in the topnote
129 0256 1 file, and indexing records get sent to the indexing routines.
130 0257 1
131 0258 1 FORMAL PARAMETERS:
132 0259 1
133 0260 1 justify - Indicates whether or not the line should be justified.
134 0261 1
135 0262 1 IMPLICIT INPUTS: None
136 0263 1
137 0264 1 IMPLICIT OUTPUTS: None
138 0265 1
139 0266 1 ROUTINE VALUE:
140 0267 1 COMPLETION CODES: None
141 0268 1
142 0269 1 SIDE EFFECTS: None
143 0270 1 --
144 0271 1
145 0272 1 BEGIN
146 0273 1
147 0274 1 IF .TSF_INDEX ! Is this TSF something for the index?
148 0275 1 OR .TSF_BTC ! Is this TSF something for the table of contents?
149 0276 1 THEN
150 0277 1 BEGIN ! This record describes an index or table of contents entry
151 0278 1 LOCAL
152 0279 1 IADDR, ! Address of text (as opposed to CH$PTR)
153 0280 1 INT_HL, ! Internal length of text.
154 0281 1 IPTR; ! CH$PTR to the text.
155 0282 1
156 0283 1 INT_HL = .TSF_INT_HL; ! Save internal length of text.
157 0284 1 IPTR = .FS_START(MRA); ! Save pointer to text.
158 0285 1 IADDR = (.FS_START(MRA))<0,XBPADDR,0> + %BLISS(BLISS36);
159 0286 1
160 0287 1 TSF_LINES = 0; ! This does not generate any text.
161 0288 1
162 0289 1 %IF DSRPLUS %THEN
163 0290 1
164 0291 1 Direct indexing and table of contents information from the topnote
165 0292 1 to the topnote file. Such information from the main body of the
166 0293 1 document goes directly to the processing routine.
167 0294 1
168 0295 1 IF .TN_COLLECTING ! If collecting topnotes,
169 0296 1 THEN
170 0297 1 TNFIL (TN_WRITE) ! save information in topnote file.
171 0298 1 ELSE
172 0299 1 BEGIN
173 0300 1 %FI
174 0301 1
175 0302 1
```



```
176 0303      ! Direct indexing and table of contents information from the footnote
177 0304      ! to the footnote file. Such information from the main body of the
178 0305      ! document goes directly to the processing routine.
179 0306
180 0307      IF .FNCT_COLLECTING      ! If collecting footnotes,
181 0308      THEN
182 0309          FOOFIL (FOO_WRIT)      ! save information in footnote file.
183 0310      ELSE
184 0311          BEGIN
185 0312              ! If user wants to debug the index or table of contents,
186 0313              ! regurgitate the information.
187 0314              IF (NOT .GCA_SKIP_OUT)
188 0315              THEN
189 0316                  BEGIN
190 0317                      ! User wants to see information on this document page.
191 0318                      IF .GCA_DEBUG_INDEX
192 0319                      AND .TSF_INDEX
193 0320                      THEN
194 0321                          ! Echo an index entry.
195 0322                          UNPUS (0)
196 0323                      ELSE
197 0324                          BEGIN
198 0325                              IF .GCA_DEBUG TOC
199 0326                              AND .TSF_BTC
200 0327                              THEN
201 0328                                  ! Echo a table of contents entry.
202 0329                                  UNPUS (1);
203 0330                              END;
204 0331                          END;
205 0332                      END;
206 0333
207 0334      ! Send index entries to the intermediate file, only if the user
208 0335      ! said /INTERMEDIATE.
209 0336      IF NOT .GCA_BIX
210 0337      THEN
211 0338          IF .TSF_INDEX
212 0339          THEN
213 0340              XOUT (.INT_HL, .IPTR, .TSF_FIRST_XTN, .TSF_H_BARS);
214 0341
215 0342      IF (.gca_bix AND .tsf_index)
216 0343      THEN
217 0344          putndy (.int_hl, .iaddr, .tsf_first_xtn, .tsf_h_bars);
218 0345
219 0346      ! Send table of contents records to the intermediate file.
220 0347      IF .tsf_btc
221 0348      THEN
222 0349          puttxt (.int_hl, .iptr, .tsf_major, .tsf_minor);
223 0350
224 0351      END;
225 0352
226 0353      %IF DSRPLUS %THEN
227 0354      %FI
228 0355      END;
229 0356
230 0357      U
231 0358      U
232 0359
```

```
233 0360
234 0361 RETURN;
235 0362 END;
236 0363
237 0364
238 0365
239 0366
240 0367
241 U 0368
242 U 0369
243 U 0370
244 U 0371
245 U 0372
246 U 0373
247 U 0374
248 U 0375
249 U 0376
250 U 0377
251 U 0378
252 U 0379
253 U 0380
254 U 0381
255 U 0382
256 U 0383
257 U 0384
258 U 0385
259 U 0386
260 U 0387
261 U 0388
262 U 0389
263 U 0390
264 U 0391
265 U 0392
266 U 0393
267 U 0394
268 U 0395
269 U 0396
270 U 0397
271 U 0398
272 U 0399
273 U 0400
274 U 0401
275 U 0402
276 U 0403
277 U 0404
278 U 0405
279 U 0406
280 U 0407
281 U 0408
282 U 0409
283 U 0410
284 U 0411
285 U 0412
286 U 0413
287 U 0414
288 U 0415
289 U 0416

      RETURN;
      END;

      +
      - This is a record full of "normal" text to be output.

      XIF DSRPLUS XTHEN
      IF NOT .TN_EXPANDING ! Expanding topnotes?
      THEN
      BEGIN
      XFI
      IF NOT .FNCT_EXPANDING ! Expanding footnotes?
      THEN
      BEGIN
      Set up justification for all lines except those that are
      being fetched from the footnote file. For these records,
      the information was computed before they were written to
      the footnote temporary file.
      BEGIN
      TSF_JUSTIFY = .JUSTIFY;
      TSF_JUST_ALG = .TSF_JUST_ALG + 1; ! Justification algorithm feedback for next time.
      TSF_PADDING = .SCA_RM - .TSF_EXT_HL; ! Space count for padding.
      END;
      XIF DSRPLUS XTHEN
      END;
      IF NOT .TN_COLLECTING
      THEN
      BEGIN
      XFI
      IF .TSF_PADDING LSS 0 AND NOT .FNCT_COLLECTING
      THEN
      BEGIN
      This can happen only if a word is encountered that does
      not fit onto the line between the margins. It can also
      happen as a side effect of breaking a word that does not
      fit in TSF (see ENDCHR, where this is done).
      The effect of the error handling is that the line is
      output without justification, but with nothing removed.
      The resulting line may have text exceeding the right margin.
      ERM (RNFCJL, 0, 0);
      XIF DSRPLUS XTHEN
      IF NOT .TN_EXPANDING ! Expanding topnotes?
      THEN
      BEGIN
      XFI
      IF NOT .FNCT_EXPANDING ! Expanding footnotes?
      THEN
      Output line and page only if this is not happening
      inside a footnote. If it's happening inside a footnote
      then the line and page numbers will be the wrong ones.
```



```
290      0417      ERM (RNFLOC, 0, 0);
291      0418
292      0419      TSF_PADDING = 0;
293      0420
294      U 0421      %IF DSRPLUS %THEN
295      U 0422      END;
296      0423      %FI
297      0424
298      0425      END;
299      0426
300      U 0427      %IF DSRPLUS %THEN
301      U 0428      END;
302      U 0429
303      U 0430      IF NOT .TN_COLLECTING      ! Collecting topnotes?
304      U 0431      THEN
305      U 0432      BEGIN
306      U 0433      %FI
307      0434
308      0435      IF NOT .FNCT_COLLECTING      ! Collecting footnotes?
309      0436      THEN
310      0437      ! Output some text directly to the document.
311      0438      BEGIN
312      0439      FS_INIT (FRA);
313      0440
314      U 0441      %IF DSRPLUS %THEN
315      U 0442      GCA_LINE_PEND = 0;      ! This line is going to be output, pend no more.
316      0443      %FI
317      0444
318      0445      LOUT ();
319      0446      END
320      0447      ELSE
321      0448      ! Save this record in the footnote file for use later.
322      0449      BEGIN
323      0450
324      0451      ! Compute the number of lines represented by this record.
325      0452      ! TSF_LINES currently contains a count of the number of
326      0453      ! lines that will be generated when the code generated by
327      0454      ! the Gxxxx routines (see module GCODE) is 'executed'.
328      0455      IF .TSF_UND AND .OUTOPT_UND_SEP
329      0456      THEN
330      0457      ! Add an extra line if underlining is done by putting something
331      0458      ! on an extra line (e.g. dashes under the text).
332      0459      TSF_LINES = .TSF_LINES + 1;
333      0460
334      0461      IF .TSF_EXT_HL NEQ 0
335      0462      THEN
336      0463      TSF_LINES = .TSF_LINES + 1;
337      0464
338      0465      ! Now actually save the record.
339      0466      FOOFIL (FOO_WRIT);
340      0467      END;
341      0468
342      U 0469      %IF DSRPLUS %THEN
343      U 0470      END
344      U 0471      ELSE
345      U 0472      BEGIN
346      U 0473
```

```
347 U 0474 2 IF .TSF_UND AND .OUTOPT_UND_SEP
348 U 0475 THEN
349 U 0476 TSF_LINES = .TSF_LINES + 1;
350 U 0477
351 U 0478 IF .TSF_EXT_HL NEQ 0
352 U 0479 THEN
353 U 0480 TSF_LINES = .TSF_LINES + 1;
354 U 0481
355 U 0482 TNFIL (TN_WRITE);
356 U 0483 END;
357 U 0484
358 U 0485 IF NOT .TN_EXPANDING ! Expanding topnotes?
359 U 0486 THEN
360 U 0487 BEGIN
361 U 0488 ZFI
362 U 0489
363 U 0490 IF NOT .FNCT_EXPANDING ! Expanding footnotes?
364 U 0491 THEN
365 U 0492 ! Throw away leftovers from line just output.
366 U 0493 ! Note that it wouldn't hurt anything to always do this. It's just that
367 U 0494 ! it's a waste of time if footnotes are being output.
368 U 0495 BEGIN
369 U 0496 TSF_INT_HL = 0;
370 U 0497 TSF_EXT_HL = 0;
371 U 0498 TSF_INT_VL = 0;
372 U 0499 TSF_NBITS = 0;
373 U 0500 TSF_BARS = FALSE; ! Turn off change bars.
374 U 0501 TSF_H_BARS = FALSE;
375 U 0502 TSF_JOS_CNT = 0;
376 U 0503 TSF_FIRST_XTN = 0;
377 U 0504 TSF_LAST_XTN = 0;
378 U 0505 SCA_WRD_LST_JUS = 0;
379 U 0506 SCA_WRD_LST_UND = FALSE;
380 U 0507 SCA_WRD_LST_SP = 0;
381 U 0508 TSF_TEXT = FALSE; ! There's nothing there now.
382 U 0509 TSF_ADJUST = 0;
383 U 0510 TSF_NEXT_REG = 0;
384 U 0511 TSF_LINES = 0;
385 U 0512 TSF_FOOTW = 0; ! No footnotes attached to this line.
386 U 0513 ! Clean up the MRA by moving the last word in the buffer to the left.
387 U 0514 SCL ();
388 U 0515 END;
389 U 0516
390 U 0517 ZIF DSRPLUS ZTHEN
391 U 0518 END;
392 U 0519 ZFI
393 U 0520
394 U 0521 1 END; ! End of OUTLIN
```

.TITLE OUTLIN directs output or move of TSF  
.IDENT \V04-000\

.EXTRN BRNOOB, FOOREC, FNCT  
.EXTRN FRA, GCA, MRA, OUTOPT  
.EXTRN SCA, TSF, RNFCLJL  
.EXTRN RNFLOC, ENDWRD, ERM

				OFFC 00000										
				5B	00000000G	EF	9E	00002	MOVAB	ERM, R11				
				5A	00000000G	EF	9E	00009	MOVAB	FOOFIL, R10				
				59	00000000G	EF	9E	00010	MOVAB	GCA+116, R9				
				58	00000000G	EF	9E	00017	MOVAB	FRA, R8				
				57	00000000G	EF	9E	0001E	MOVAB	FNCT+20, R7				
				56	00000000G	EF	9E	00025	MOVAB	TSF, R6				
				52		66	D0	0002C	MOVL	TSF, R2		0274		
				04	14	A2	E8	0002F	BLBS	20(R2), 1\$				
				75	2C	A2	E9	00033	BLBC	44(R2), 8\$		0275		
				53		62	D0	00037	1\$:	MOVL	(R2), INT HL	0283		
				54	00000000G	FF	DC	0003A	MOVL	@MRA, IPTR		0284		
				55	00000000G	FF	D0	00041	MOVL	@MRA, IADDR		0285		
					34	A2	D4	00048	CLRL	52(R2)		0287		
				06		67	E9	0004B	BLBC	FNCT+20, 2\$		0307		
						04	DD	0004E	PUSHL	#4		0309		
				6A		01	FB	00050	CALLS	#1, FOOFIL				
							04	00053	RET					
				1C	FC	A9	E8	00054	2\$:	BLBS	GCA+112, 5\$	0315		
				08		69	E9	00058	BLBC	GCA+116, 3\$		0320		
				04	14	A2	E9	0005B	BLBC	20(R2), 3\$		0321		
						7E	D4	0005F	CLRL	-(SP)		0324		
						0A	11	00061	BRB	4\$				
				0D		01	E1	00063	3\$:	BBC	#1, GCA+116, 5\$	0328		
						09	2C	A2	E9	00067	BLBC	44(R2), 5\$	0329	
							01	DD	0006B	PUSHL	#1	0332		
					00000000G	EF	01	FB	0006D	4\$:	CALLS	#1, UNPUS		
				1D	08	A9	02	E1	00074	5\$:	BBC	#2, GCA+124, 6\$	0346	
						50	66	D0	00079	MOVL	TSF, R0			
						16	14	A0	E9	0007C	BLBC	20(R0), 6\$		
						50	66	D0	00080	MOVL	TSF, R0		0348	
7E	0080	C0				01	00	EF	00083	EXTZV	#0, #1, 128(R0), -(SP)			
							38	A0	DD	0008A	PUSHL	56(R0)		
							28	BB	0008D	PUSHR	#*M<R3,R5>			
					00000000G	EF	04	FB	0008F	CALLS	#4, PUTNDY			
						50	66	D0	00096	6\$:	MOVL	TSF, R0	0351	
						01	2C	A0	E8	00099	BLBS	44(R0), 7\$		
								04	0009D	RET				
						7E	38	A0	7D	0009E	7\$:	MOVQ	56(R0), -(SP)	0353
							18	BB	000A2	PUSHR	#*M<R3,R4>			
					00000000G	EF	04	FB	000A4	CALLS	#4, PUTTXT			
								04	000AB	RET			0277	
						15	04	A7	E8	000AC	8\$:	BLBS	FNCT+24, 9\$	0374
						50	66	D0	000B0	MOVL	TSF, R0		0380	
					24	A0	04	AC	D0	000B3	MOVL	JUSTIFY, 36(R0)		0381
						64	A0	D6	000B8	INCL	100(R0)		0382	
							04	A0	C3	000BB	SUBL3	4(R0), @SCA+120, 64(R0)		0383
40	A0	00000000G				50	66	D0	000C5	9\$:	MOVL	TSF, R0		0394
							40	A0	D5	000C8	TSTL	64(R0)		
							23	18	000CB	BGEQ	11\$			
						37	67	E8	000CD	BLBS	FNCT+20, 12\$			



OUTLIN  
V04-000

directs output or move of TSF  
OUTLIN -- output full MRA.

E 13  
16-Sep-1984 01:22:27 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 13:07:33 [RUNOFF.SRC]OUTLIN.BLI;1

Page 10  
(4)

			7E	7C	000D0	CLRQ	-(SP)	0404	
		00000000G	8F	DD	000D2	PUSHL	#RNF C JL		
6B			03	FB	000D8	CALLS	#3, ERM		
0B		04	A7	E8	000DB	BLBS	FNCT+24, 10\$	0412	
			7E	7C	000DF	CLRQ	-(SP)	0417	
		00000000G	8F	DD	000E1	PUSHL	#RNFLOC		
6B			03	FB	000E7	CALLS	#3, ERM		
50			66	D0	000EA	10\$: MOVL	TSF, R0		
		40	A0	D4	000ED	CLRL	64(R0)	0419	
14			67	E8	000F0	11\$: BLBS	FNCT+20, 12\$	0435	
		0C	A8	D4	000F3	CLRL	FRA+12	0439	
6B		10	A8	9E	000F6	MOVAB	FRA+16, FRA		
04	04		68	D0	000FA	MOVL	FRA, FRA+4		
00000000G	EF		00	FB	000FE	CALLS	#0, LOUT	0445	
			1F	11	00105	BRB	15\$	0435	
50			66	D0	00107	12\$: MOVL	TSF, R0	0455	
0A	08		01	E1	0010A	BBC	#1, 8(R0), 13\$		
		03	EF	E9	0010F	BLBC	OUTOPT+8, 13\$		
			34	A0	D6	00116	INCL	52(R0)	0459
			04	A0	D5	00119	13\$: TSTL	4(R0)	0461
			03	13	0011C	BEQL	14\$		
			34	A0	D6	0011E	INCL	52(R0)	0463
			04	DD	00121	14\$: PUSHL	#4	0466	
6A			01	FB	00123	CALLS	#1, FOOFIL		
3A		04	A7	E8	00126	15\$: BLBS	FNCT+24, 16\$	0490	
50			66	D0	0012A	MOVL	TSF, R0	0495	
			60	7C	0012D	CLRQ	(R0)	0496	
		18	A0	D4	0012F	CLRL	24(R0)	0498	
7C	A0		01	8A	00132	BICB2	#1, 124(R0)	0500	
0080	CO		01	8A	00136	BICB2	#1, 128(R0)	0501	
		20	A0	D4	0013B	CLRL	32(R0)	0502	
		38	A0	7C	0013E	CLRQ	56(R0)	0503	
		00000000G	EF	7C	00141	CLRQ	SCA+336	0505	
		00000000G	EF	D4	00147	CLRL	SCA+332	0507	
		60	A0	D4	0014D	CLRL	96(R0)	0508	
		28	A0	D4	00150	CLRL	40(R0)	0509	
		0088	CO	D4	00153	CLRL	136(R0)	0510	
		34	A0	D4	00157	CLRL	52(R0)	0511	
		08	A0	7C	0015A	CLRQ	8(R0)	0499	
00000000G	EF		00	FB	0015D	CALLS	#0, SCL	0514	
			04	00164	16\$: RET			0521	

: Routine Size: 357 bytes, Routine Base: \$CODE\$ + 0000

: 395 0522 1

```
397 0523 1 %sbttl 'OUTNJ -- output text unjustified'
398 0524 1 GLOBAL ROUTINE outnj : NOVALUE =
399 0525 1
400 0526 1 ++
401 0527 1 FUNCTIONAL DESCRIPTION:
402 0528 1
403 0529 1 Forces out the current text, without allowing it to be justified.
404 0530 1
405 0531 1 FORMAL PARAMETERS: None
406 0532 1
407 0533 1 IMPLICIT INPUTS: None
408 0534 1
409 0535 1 IMPLICIT OUTPUTS: None
410 0536 1
411 0537 1 ROUTINE VALUE:
412 0538 1 COMPLETION CODES: None
413 0539 1
414 0540 1 SIDE EFFECTS: None
415 0541 1
416 0542 1 --
417 0543 1
418 0544 1 BEGIN
419 0545 1
420 0546 1 IF ( NOT .SCA_FC)
421 0547 1 OR ( NOT .SCA_FILL)
422 0548 1 THEN
423 0549 1 ENDWRD (FALSE, FALSE, FALSE)
424 0550 1 ELSE
425 0551 1 BEGIN
426 0552 1
427 0553 1 IF (.SCA_WRD_FOOTW NEQ 0) ! Footnotes attached to this word?
428 0554 1 THEN
429 0555 1 ! Take care of pending footnote.
430 0556 1 BEGIN
431 0557 1 TSF_FOOTW = .TSF_FOOTW + .SCA_WRD_FOOTW; ! Update count of footnotes associated with this lin
432 0558 1 SCA_WRD_FOOTW = 0 ! Make sure footnotes don't get counted twice
433 0559 1 END;
434 0560 1
435 0561 1 ! is there an index entry associated with this word?
436 0562 1 IF .SCA_WRD_F_XTN NEQ 0
437 0563 1 THEN
438 0564 1 ! Take care of pending index entry.
439 0565 1 BEGIN
440 0566 1
441 0567 1 IF .TSF_FIRST_XTN EQL 0
442 0568 1 THEN
443 0569 1 TSF_FIRST_XTN = .SCA_WRD_F_XTN;
444 0570 1
445 0571 1 TSF_LAST_XTN = .SCA_WRD_L_XTN;
446 0572 1 END;
447 0573 1 END;
448 0574 1
449 0575 1 SCA_WRD_F_XTN = 0;
450 0576 1 SCA_WRD_L_XTN = 0;
451 0577 1 OUTLIN (FALSE); ! Don't justify line.
452 0578 1 TSF_JUST_ALG = 0; ! Reset justification algorithm.
453 0579 1 SCA_CONT = FALSE; ! Don't allow concatenation (.NO SPACE)
```

OUTLIN  
V04-000

directs output or move of TSF  
OUTNJ -- output text unjustified

G 13  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI;1

Page 12  
(5)

: 454

0580 1 END;

! End of OUTNJ

			000C	00000	.ENTRY	OUTNJ, Save R2,R3	0524
53	00000000G	EF	9E	00002	MOVAB	TSF, R3	
52	00000000G	EF	9E	00009	MOVAB	SCA+292, R2	
05	FF70	C2	E9	00010	BLBC	SCA+148, 1\$	0546
0D	FF44	D2	E8	00015	BLBS	SCA+104, 2\$	0547
		7E	7C	0001A	CLRQ	-(SP)	0549
		7E	D4	0001C	CLRL	-(SP)	
00000000G	EF	03	FB	0001E	CALLS	#3, ENDWRD	
		23	11	00023	BRB	5\$	
51		62	D0	00027	MOVL	SCA+292, R1	0553
		09	13	0002A	BEQL	3\$	
50		63	D0	0002C	MOVL	TSF, R0	0556
0C	A0	51	C0	0002F	ADDL2	R1, 12(R0)	0557
		62	D4	00033	CLRL	SCA+292	0558
51	04	A2	D0	00035	MOVL	SCA+296, R1	0562
		11	13	00039	BEQL	5\$	
50		63	D0	0003B	MOVL	TSF, R0	0567
	38	A0	D5	0003E	TSTL	56(R0)	
		04	12	00041	BNEQ	4\$	
38	A0	51	D0	00043	MOVL	R1, 56(R0)	0569
3C	A0	A2	D0	00047	MOVL	SCA+300, 60(R0)	0571
		04	A2	7C	0004C	CLRQ	SCA+296
		7E	D4	0004F	CLRL	-(SP)	0577
FE45	CF	01	FB	00051	CALLS	#1, OUTLIN	
	50	63	D0	00056	MOVL	TSF, R0	
		64	A0	D4	00059	CLRL	100(R0)
		80	A2	D4	0005C	CLRL	SCA+164
			04	0005F	RET		0578
							0579
							0580

: Routine Size: 96 bytes, Routine Base: \$CODE\$ + 0165

: 455

0581 1



```
457 0582 1 %sbttl 'OUTJ -- output text justified'
458 0583 1 GLOBAL ROUTINE outj : NOVALUE =
459 0584 1
460 0585 1 ++
461 0586 1 FUNCTIONAL DESCRIPTION:
462 0587 1
463 0588 1 Forces out the current text, causing it to be justified.
464 0589 1
465 0590 1 FORMAL PARAMETERS: None
466 0591 1
467 0592 1 IMPLICIT INPUTS: None
468 0593 1
469 0594 1 IMPLICIT OUTPUTS: None
470 0595 1
471 0596 1 ROUTINE VALUE:
472 0597 1 COMPLETION CODES: None
473 0598 1
474 0599 1 SIDE EFFECTS: None
475 0600 1
476 0601 1 --
477 0602 1
478 0603 2 BEGIN
479 0604 2
480 0605 2 IF ( NOT .SCA_FC)
481 0606 2 OR ( NOT .SCA_FILL)
482 0607 2 THEN
483 0608 2 ENDWRD (FALSE, FALSE, FALSE)
484 0609 2 ELSE
485 0610 2 BEGIN
486 0611 2
487 0612 2 IF (.SCA_WRD_FOOTW NEQ 0) ! Footnotes attached to this word?
488 0613 2 THEN
489 0614 2 ! Take care of pending footnote.
490 0615 2 BEGIN
491 0616 2 TSF_FOOTW = .TSF_FOOTW + .SCA_WRD_FOOTW; ! Update count of footnotes associated with this lin
492 0617 2 SCA_WRD_FOOTW = 0 ! Make sure footnotes are not counted twice
493 0618 2 END;
494 0619 2
495 0620 2 IF .SCA_WRD_F_XTN NEQ 0
496 0621 2 THEN
497 0622 2 ! Take care of pending index entry.
498 0623 2 BEGIN
499 0624 2
500 0625 2 IF .TSF_FIRST_XTN EQL 0
501 0626 2 THEN
502 0627 2 TSF_FIRST_XTN = .SCA_WRD_F_XTN;
503 0628 2
504 0629 2 TSF_LAST_XTN = .SCA_WRD_L_XTN;
505 0630 2 END;
506 0631 2 END;
507 0632 2
508 0633 2 SCA_WRD_F_XTN = 0;
509 0634 2 SCA_WRD_L_XTN = 0;
510 0635 2 OUT[IN TRUE);
511 0636 2 TSF_JUST_ALG = 0;
512 0637 2 SCA_CONT = FALSE;
513 0638 2 END;

! Justify line.
! Reset justification algorithm.
! Don't allow concatenation (.NO SPACE)
! End of OUTJ
```

			000C	00000	.ENTRY	OUTJ, Save R2,R3	0583
53	00000000G	EF	9E	00002	MOVAB	TSF, R3	
52	00000000G	EF	9E	00009	MOVAB	SCA+292, R2	
05	FF70	C2	E9	00010	BLBC	SCA+148, 1\$	0605
0D	FF44	D2	E8	00015	BLBS	@SCA+104, 2\$	0606
		7E	7C	0001A	CLRQ	-(SP)	0608
		7E	D4	0001C	CLRL	-(SP)	
00000000G	EF	03	FB	0001E	CALLS	#3, ENDWRD	
		25	11	00025	BRB	5\$	
51		62	D0	00027	MOVL	SCA+292, R1	0612
		09	13	0002A	BEQL	3\$	
50		63	D0	0002C	MOVL	TSF, R0	0615
0C	A0	51	C0	0002F	ADDL2	R1, 12(R0)	0616
		62	D4	00033	CLRL	SCA+292	0617
51	04	A2	D0	00035	MOVL	SCA+296, R1	0620
		11	13	00039	BEQL	5\$	
50		63	D0	0003B	MOVL	TSF, R0	0625
	38	A0	D5	0003E	TSTL	56(R0)	
		04	12	00041	BNEQ	4\$	
38	A0	51	D0	00043	MOVL	R1, 56(R0)	0627
3C	A0	08	A2	00047	MOVL	SCA+300, 60(R0)	0629
		04	A2	0004C	CLRQ	SCA+296	0633
		01	DD	0004F	PUSHL	#1	0635
FDE5	CF	01	FB	00051	CALLS	#1, OUTLIN	
	50	63	D0	00056	MOVL	TSF, R0	
		64	A0	00059	CLRL	100(R0)	0636
		80	A2	0005C	CLRL	SCA+164	0637
			04	0005F	RET		0638

; Routine Size: 96 bytes. Routine Base: \$CODE\$ + 01C5

; 514 0639 1

```
0640 1 %sbttl 'OUTCRG -- remove carriage control from TSF'
0641 1 GLOBAL ROUTINE outcr: NOVALUE =
0642 1
0643 1 ++
0644 1 FUNCTIONAL DESCRIPTION:
0645 1
0646 1     Clears out carriage control sequences from the TSF.
0647 1
0648 1 FORMAL PARAMETERS:      None
0649 1
0650 1 IMPLICIT INPUTS:        None
0651 1
0652 1 IMPLICIT OUTPUTS:       None
0653 1
0654 1 ROUTINE VALUE:
0655 1 COMPLETION CODES:       None
0656 1
0657 1 SIDE EFFECTS:           None
0658 1 --
0659 1
0660 2 BEGIN
0661 2 ! This is a record full of 'normal' text to be output.
0662 2 FS_INIT (FRA);
0663 2 TSF_JUSTIFY = FALSE;
0664 2 TSF_PADDING = 0;
0665 2
0666 2 U XIF DSRPLUS %THEN
0667 2
0668 2 ! If collecting topnotes, go through OUTNJ to make sure blank lines
0669 2 ! get put in the right place.
0670 2 U IF .TN_COLLECTING ! Collecting topnotes?
0671 2 U THEN
0672 2 U     OUTNJ ()
0673 2 U ELSE
0674 2 U     BEGIN
0675 2 U XFI
0676 2 U
0677 2 U ! If collecting footnotes, go through OUTNJ to make sure blank lines
0678 2 U ! get put in the right place.
0679 2 U IF .FNCT_COLLECTING ! Collecting footnotes?
0680 2 U THEN
0681 2 U     OUTNJ ()
0682 2 U ELSE
0683 2 U     LOUT ();
0684 2 U
0685 2 U XIF DSRPLUS %THEN
0686 2 U     END;
0687 2 U XFI
0688 2 U
0689 2 U ! Throw away leftovers from line just output.
0690 2 U TSF_INT_HL = 0;
0691 2 U TSF_EXT_HL = 0;
0692 2 U TSF_INT_VL = 0;
0693 2 U TSF_NBITS = 0;
0694 2 U TSF_BARS = FALSE;
0695 2 U TSF_H_BARS = FALSE;
0696 2 U TSF_JDS_CNT = 0;
0697 2 U ! Turn off change bars.
```



OUTLIN  
V04-000

directs output or move of TSF  
OUTCRG -- remove carriage control from TSF

K 13  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI;1

Page 16  
(7)

```

: 573      0697 2    TSF_FIRST_XTN = 0;
: 574      0698 2    TSF_LAST_XTN = 0;
: 575      0699 2    TSF_TEXT = FALSE;
: 576      0700 2    TSF_ADJUST = 0;
: 577      0701 2    TSF_FOOTW = 0;
: 578      0702 2    FS_INIT (MRA);
: 579      0703 2    SCA_WRD_PNTR = FS_START (MRA);
: 580      0704 2    TSF_NEXT_REG = 0;
: 581      0705 1    END;
```

! There's nothing there now.

! End of OUTCRG

			000C 00000	.ENTRY OUTCRG, Save R2,R3	0641
	53	00000000G	EF 9E 00002	MOVAB TSF, R3	
	52	00000000G	EF 9E 00009	MOVAB FRA, R2	
		0C	A2 D4 00010	CLRL FRA+12	0662
	62	10	A2 9E 00013	MOVAB FRA+16, FRA	
04	A2		62 D0 00017	MOVL FRA, FRA+4	
	50		63 D0 0001B	MOVL TSF, R0	
		24	A0 D4 0001E	CLRL 36(R0)	0663
		40	A0 D4 00021	CLRL 64(R0)	0664
	07	00000000G	EF E9 00024	BLBC FNCT+20, 1\$	0679
FF10	CF		00 FB 0002B	CALLS #0, OUTNJ	0681
			07 11 00030	BRB 2\$	
00000000G	EF		00 FB 00032 1\$:	CALLS #0, LOUT	0683
	50		63 D0 00039 2\$:	MOVL TSF, R0	0686
			60 7C 0003C	CLRL (R0)	0690
		18	A0 D4 0003E	CLRL 24(R0)	0692
	7C	A0	01 8A 00041	BICB2 #1, 124(R0)	0694
0080	C0		01 8A 00045	BICB2 #1, 128(R0)	0695
		20	A0 D4 0004A	CLRL 32(R0)	0696
		38	A0 7C 0004D	CLRL 56(R0)	0697
		60	A0 D4 00050	CLRL 96(R0)	0699
		28	A0 D4 00053	CLRL 40(R0)	0700
		08	A0 7C 00056	CLRL 8(R0)	0693
	51	00000000G	EF D0 00059	MOVL MRA, R1	0702
		0C	A1 D4 00060	CLRL 12(R1)	
	61	10	A1 9E 00063	MOVAB 16(R1), (R1)	
04	A1		61 D0 00067	MOVL (R1), 4(R1)	
00000000G	EF		61 D0 0006B	MOVL (R1), SCA+248	0703
		0088	C0 D4 00072	CLRL 136(R0)	0704
			04 00076	RET	0705

; Routine Size: 119 bytes, Routine Base: \$CODE\$ + 0225

; 582 0706 1

```
584 0707 1 %sbttl 'OUTPAS -- directs pass-through record'
585 0708 1 GLOBAL ROUTINE outpas (fullwords, address, xtn_pointer, type) : NOVALUE =
586 0709 1
587 0710 1 ++
588 0711 1 FUNCTIONAL DESCRIPTION:
589 0712 1
590 0713 1     Directs a passthrough record to the appropriate location.
591 0714 1
592 0715 1 FORMAL PARAMETERS:
593 0716 1
594 0717 1     fullwords    - Number of fullwords in the vector pointed to by address.
595 0718 1     address      - Address of the block of data to write.
596 0719 1     type         - Minor record type, as defined in FOOREC.REQ or TNREC.REQ.
597 0720 1     xtn_pointer  - Offset into the passthrough record to apply in
598 0721 1                  order to find the transaction number.
599 0722 1
600 0723 1 IMPLICIT INPUTS:      None
601 0724 1
602 0725 1 IMPLICIT OUTPUTS:     None
603 0726 1
604 0727 1 ROUTINE VALUE:
605 0728 1 COMPLETION CODES:      None
606 0729 1
607 0730 1 SIDE EFFECTS:          None
608 0731 1 --
609 0732 1
610 0733 1 BEGIN
611 0734 1
612 U 0735 1 %IF DSRPLUS %THEN
613 UU 0736 1
614 UU 0737 1     If topnotes are being collected, then write the record to the topnote
615 UU 0738 1     file. Otherwise, direct the record to the appropriate utility.
616 UU 0739 1
617 UU 0740 1     IF .TN_COLLECTING
618 UU 0741 1     THEN
619 UU 0742 1
620 UU 0743 1         Set up some fields in the record header and
621 UU 0744 1         write the record to the work file.
622 UU 0745 1
623 UU 0746 1         BEGIN
624 UU 0747 1         TNREC_MAJOR_TYPE = TNREC_MAJ_PASS;
625 UU 0748 1         TNREC_MINOR_TYPE = .TYPE;
626 UU 0749 1         TNREC_XTN_PTR   = .XTN_POINTER;
627 UU 0750 1         TNREC_ADDRESS  = .ADDRESS;
628 UU 0751 1         TNREC_RECORD_SIZE = .FULLWORDS;
629 UU 0752 1         TNFIL (TN_PWRITE)
630 UU 0753 1         END
631 UU 0754 1     ELSE
632 U 0755 1     BEGIN
633 0756 1 %FI
634 0757 1
635 0758 1     ! If footnotes are being collected, then write the record to the footnote
636 0759 1     ! work file. Otherwise, direct the record to the appropriate utility.
637 0760 1     IF .FNCT_COLLECTING
638 0761 1     THEN
639 0762 1         ! We are between .FOOTNOTE and .END FOOTNOTE
640 0763 1         ! Write the record to the work file.
```

OUTLIN  
V04-000

directs output or move of TSF  
OUTPAS -- directs pass-through record

M 13  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI;1

Page 18  
(8)

```

: 641      0764      BEGIN
: 642      0765      ! First set up some fields in the record header.
: 643      0766      FOOREC_MAJOR_TYPE = FOOREC_MAJ_PASS;
: 644      0767      FOOREC_MINOR_TYPE = .TYPE;
: 645      0768      FOOREC_XTN_PTR = .XTN_POINTER;
: 646      0769      FOOREC_ADDRESS = .ADDRESS;
: 647      0770      FOOREC_RECORD_SIZE = .FULLWORDS;
: 648      0771      FOOFIL (FOO_PWRIT)
: 649      0772      END
: 650      0773      ELSE
: 651      0774      ! Not between .FOOTNOTE and .END FOOTNOTE. Direct the record to
: 652      0775      ! the appropriate .BXX file and write the binary record.
: 653      0776      $XPO_PUT (IOB = BRNOOB, BINARY_DATA = (.FULLWORDS, .ADDRESS));
: 654      0777
: 655      U 0778      %IF DSRPLUS %THEN
: 656      U 0779      END;
: 657      0780      %FI
: 658      0781
: 659      0782      END;

```

! End of OUTPAS

```

                                .EXTRN  XPOS$PUT, XPOS$FAILURE
                                .ENTRY   OUTPAS, Save R2,R3
                                MOVAB    IOB$+68, R3
                                MOVAB    FOOREC, R2
                                SUBL2    #8, SP
                                BLBC     FNCT+20, 1$
                                MOVL     #2, FOOREC
                                MOVL     TYPE, FOOREC+4
                                MOVL     XTN_POINTER, FOOREC+12
                                MOVL     ADDRESS, FOOREC+16
                                MOVL     FULLWORDS, FOOREC+8
                                PUSHL    #10
                                CALLS    #1, FOOFIL
                                RET
                                MULW3    #4, FULLWORDS, $IOB$OUTPUT
                                MOVB     #2, $IOB$OUTPUT+2
                                MOVB     #1, $IOB$OUTPUT+3
                                MOVL     ADDRESS, $IOB$OUTPUT+4
                                MOVAB    $IOB$OUTPUT, IOB$+68
                                MOVB     #7, IOB$+44
                                PUSHAB   XPOS$FAILURE
                                CLRL     -(SP)
                                PUSHAB   IOB$
                                CALLS    #3, XPOS$PUT
                                RET

```

0708

0760

0766

0767

0768

0769

0770

0771

0776

0782

; Routine Size: 103 bytes, Routine Base: \$CODE\$ + 029C

```

: 660      0783      1
: 661      0784      1 END
: 662      0785      0 ELUDOM

```

! End of module



OUTLIN  
V04-000

directs output or move of TSF  
OUTPAS -- directs pass-through record

N 13  
16-Sep-1984 01:22:27  
14-Sep-1984 13:07:33

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]OUTLIN.BLI:1

Page 19  
(8)

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	771	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	99	16	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	73	5	86	00:00.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:OUTLIN/OBJ=OBJ\$:OUTLIN MSRC\$:OUTLIN/UPDATE=(ENHS:OUTLIN)

: Size: 771 code + 0 data bytes  
: Run Time: 00:18.6  
: Elapsed Time: 00:39.1  
: Lines/CPU Min: 2529  
: Lexemes/CPU-Min: 23474  
: Memory Used: 156 pages  
: Compilation Complete



0346 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

NEWPG LIS	NODOPX LIS	OFT LIS	OUTXT LIS
NDXURS LIS	NOTE LIS	OUTLIN LIS	PACK LIS
NM LIS	OUTXHR LIS	OUTCHA LIS	OUTHOR LIS
NOXXTN LIS			